SANDIA REPORT

SAND2009-7777
Unlimited Release
Printed February 2010

Summary of the CSRI Workshop on Combinatorial Algebraic Topology (CAT): Software, Applications, & Algorithms

Janine C. Bennett, David M. Day, Scott A. Mitchell

Prepared by
Sandia National Laboratories
Albuquerque, New Mexico 87185 and Livermore, California 94550

Sandia is a multiprogram laboratory operated by Sandia Corporation,
a Lockheed Martin Company, for the United States Department of Energy's
National Nuclear Security Administration under Contract DE-AC04-94-AL85000.

Sandia National Laboratories

Issued by Sandia National Laboratories, operated for the United States Department of Energy by Sandia Corporation.

Available to DOE and DOE contractors from
U.S. Department of Energy
Office of Scientific and Technical Information
P.O. Box 62
Oak Ridge, TN 37831

Telephone: (865) 576-8401
Facsimile: (865) 576-5728
E-Mail: reports@adonis.osti.gov
Online ordering: http://www.osti.gov/bridge

Available to the public from
U.S. Department of Commerce
National Technical Information Service
5285 Port Royal Rd
Springfield, VA 22161

Telephone: (800) 553-6847
Facsimile: (703) 605-6900
E-Mail: orders@ntis.fedworld.gov
Online ordering: http://www.ntis.gov/help/ordermethods.asp?loc=7-4-0#online

SAND2009-7777
Unlimited Release
Printed February 2010

Summary of the CSRI Workshop on Combinatorial Algebraic Topology (CAT): Software, Applications, & Algorithms

Janine C. Bennett
Sandia National Laboratories
Visualization and Scientific Computing Department
P.O. Box 969, MS 9159, Livermore, CA 94551
jcbenne@sandia.gov, http://www.janinebennett.org

David M. Day
Applied Mathematics & Applications Department
P.O. Box 5800, MS 1320, Albuquerque, NM, 87185-1320
dmday@sandia.gov

Scott A. Mitchell
Sandia National Laboratories
Computer Science and Informatics Department
P.O. Box 5800, MS 1316, Albuquerque, NM, 87185-1316
samitch@sandia.gov, http://www.cs.sandia.gov/~samitch

Abstract

This report summarizes the Combinatorial Algebraic Topology: software, applications & algorithms workshop (CAT Workshop). The workshop was sponsored by the Computer Science Research Institute of Sandia National Laboratories. It was organized by CSRI staff members Scott Mitchell and Shawn Martin. It was held in Santa Fe, New Mexico, August 29-30. The CAT Workshop website has links to some of the talk slides and other information, http://www.cs.sandia.gov/CSRI/Workshops/2009/CAT/index.html.

The purpose of the report is to summarize the discussions and recap the sessions. There is a special emphasis on technical areas that are ripe for further exploration, and the plans for follow-up amongst the workshop participants. The intended audiences are the workshop participants, other researchers in the area, and the workshop sponsors.

Figure 1. The CAT Workshop logo, from http://www.cs.sandia.gov/CSRI/Workshops/2009/CAT/index.html

Acknowledgment

Thanks to the Computer Science Research Institute (CSRI) under the DOE NNSA Advanced Scientific Computing (ASC) program for sponsoring this workshop, including Sandia's CSRI program manager S. Scott Collis. Thanks to the ASC-sponsored Computer Science Research Foundation, specifically David Rogers and Suzanne Rountree, for supporting the workshop organizers and several of the Sandia participants in their research and preparation of results for this workshop.

Thanks to the many participants of the workshop who shared their knowledge, including not-yet-published results, and insights into the direction the field should take, with Sandia and their colleagues.

Contents

Figures

Nomenclature

B_k the sub-group of C_k that is the image of the boundary map, $B_k = \text{im}\,\partial_{k+1}$

β_i the ith Betti number, $\beta_i = \text{rank}\mathsf{H}_i$

C_k the group of k-dimensional chains

CAT Combinatorial Algebraic Topology

coface a coface of a simplex is one of the simplices containing it

chain a k-chain is a formal sum of k-dimensional simplices, $\sum c_i \sigma_i^k$ with $c_i \in R$ and $\sigma_i^k \in \mathbb{K}$

CT Computational Topology

cycle a chain c such that $\partial c = 0$, i.e. an element of Z

∂_k the boundary operator ∂_k for dimension k maps σ^k to its $(k-1)$-dimensional simplices with coefficients in R

face a face of a simplex is one of its sub-simplices

filtration a sequences of complexes nested by inclusion; sometimes by adding one simplex at each step of the sequence. $\emptyset \subseteq K_0 \subseteq K_1 \subseteq ... \subseteq K_m$

generator a basis element for H. A generator is a cycle.

H_i the ith homology group, $\mathsf{H}_k = Z_k/B_k$

$\mathbb{K}$ a simplicial complex

loop a cycle of edges, e.g. a rank-1 generator

$\mathbb{Q}$ the rationals

R the coefficient ring over which homology is computed. Often R is chosen to be the field $\mathbb{Z}_2$, $\mathbb{Z}_p$, or $\mathbb{Q}$; sometimes it is the ring $\mathbb{Z}$; rarely it is $\mathbb{R}$.

$\mathbb{R}$ the reals

rank the dimension of some space

σ a simplex. a d-dimensional simplex is the combination of $d+1$ vertices. Note vertices are zero-dimensional simplices.

$\mathbb{Z}_p$ the group of integers mod p, usually for p prime

Z_k the sub-group of C_k that is the kernel of the boundary map, $Z_k = \ker \partial_k$

1 Software

The Saturday morning session focused on existing software efforts for combinatorial algebraic topology. The intended audience was those wishing to use their tools for research framework or application solutions. Speakers were asked to address the following items

- scope, current and planned
- capabilities: especially Betti numbers, homology generators, generators meeting application-specific criteria filtrations & Reeb graphs for sensitivity and transients; Smith Normal Form factorization, other linear algebra capabilities; and scalability
- software maturity/usability
- availability and usage models
- speakers were asked to de-emphasize reduction, sampling, and initial complex generation

5 talks were given on Saturday morning.

- JPlex, Henry Adams (Stanford)
- The Linbox library, Clement Pernet (INRIA, Universit de Grenoble)
- An Overview of Trilinos, Heidi Thornquist (Sandia National Laboratories)
- Reeb Graphs, Attila Gyulassy (University of Utah)
- A Mathematical Tour of Geomagic Software, Yates Fletcher (Geomagic, Inc.)

See http://www.cs.sandia.gov/CSRI/Workshops/2009/CAT/program.html for links to some talks' slides.

1.1 JPlex

People

Henry Adams (speaker), Mikael Vejdemo Johansson (primary author, mik@math.stanford.edu). Afra Zomorodian (Plex).

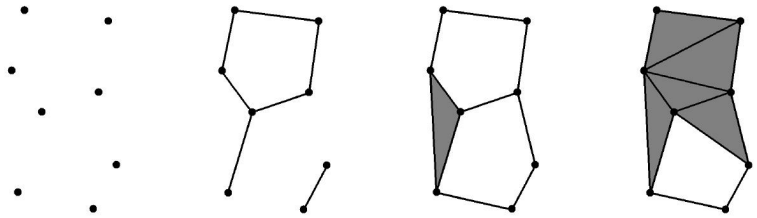

Figure 2. A filtration from a JPlex tutorial.

Summary

JPlex is the Java version of Plex. It computes persistent homology of filtered simplicial complexes using the column echelon (from Smith-Normal) form algorithm in the "Computing Persistent Homology" paper[23]. It has capabilities for defining filtrations explicitly, or generating filtered Rips complexes given vertices with either geometric locations or (possibly non-geometric) distances between them. Given a filtered simplicial complex the output is a collection of finite or semi-infinite intervals called Betti Bar codes - which are one way to describe persistent homology (persistence provides insight into when features are born and when they die). Intervals are computed by putting boundary matrices in smith normal form (column echelon form). Common applications are topological simplification, point cloud data analysis, and shape recognition.

JPLex does not currently return generators. In principle one can compute these using persistence algorithm, and it is a planned feature in the next round of development. However there are no nice geometric guarantees on the generators constructed via the persistence algorithm.

Current implementation limitations include simplex dimension < 8 and finite fields $\mathbb{Z}_p$ with $p < 256$.

There appears to be room for careful analysis of complexity for this family of algorithms in terms of dimension d, vertices v, simplices n, and/or ranks β_i. One easy bound on the algorithmic complexity the persistence algorithm is $O(n^3)$ where n is the number of simplices. However, one of the n's is in fact the dimension of complex. Also, the algorithm depends on the number of steps in the filtration. One common way to compute a filtration is to have each step of the filtration include the next-shortest edge and all the induced higher dimensional simplices. Instead, JPlex uses real-valued distance steps to build the filtration. At step j, all edges with length $< jh$ (given a fixed geometric step-size h) and all the induced higher dimensional simplices are added into the filtration. For a lot of embedded complexes the approach takes linear time in practice. For example when N is 11,151 with a length of 300 steps in the Morse Filtration, the code takes .28 seconds. Also, there is a faster $\mathbb{Z}_2$ implementation.

There were several comments to the effect that memory, not computation, is the bottleneck. In particular, just constructing the complexes can take exponential time and space. In one example 5 gigabytes of memory was used to construct 4 million 3 simplices. See also the workshop talk by

Afra Zomorodian on reducing the construction of homologically-useless simplices. Reduction and co-reduction techniques are relevant, but they work *after* the complex is already constructed.

In the new version of JPlex memory leaks have been addressed, it is much easier to install. Also, a Matlab beanshell interface is provided.

More info including source code, tutorials and a Javadoc tree can be found at http://comptop.stanford.edu/, Stanford's computational topology website[3]. JPlex has a single software developer, Mikael Vejdemo Johansson. People may be interested in the older version, Plex, as well. Some tools are not in JPlex that were in Plex2.5, such as the sequential max/min landmarks and cohomology. Plex also had tools for manipulating complexes, such as computing their intersection and union. Afra Zomorodian was involved with the development of Plex and has his own implementation currently. It appears that Plex may be more efficient in terms of speed or memory than JPlex is currently. Afra is following up with Mikael on timing studies, implementation, and complexity issues. Future versions of JPlex will re-include cohomology. It is intended that it will provide a better platform for research capabilities.

Group Discussion

Q: When addressing torsion it is nice to have higher moduli so you can understand the effects of the torsion:
A: There are constants in the code that can be changed easily to address this.

Q: What are the largest datasets handled?
A:
Comment 1: JPLex past 100,000 simplices get slow.
Comment 2: With 24 million simplices it took 63 seconds
Comment 3: Memory consumption: 5gb for 4 million simplices with JPlex (3d simplices). Zigzag persistent homology (see papers by workshop participant Dmitriy Morozov) could be used to address memory consumption- depends on number of simplices in any slice $O(nm^2)$, as one can throw away non-active loops. It looks like in current version memory (for generating the simplicies) is more of a limitation than run time. Some sort of implicit representation of simplices may be a way around this,.

Additional Opportunities

Jason Shepherd offered to host a JPlex integration with the VTK/Titan [20, 11] software framework.

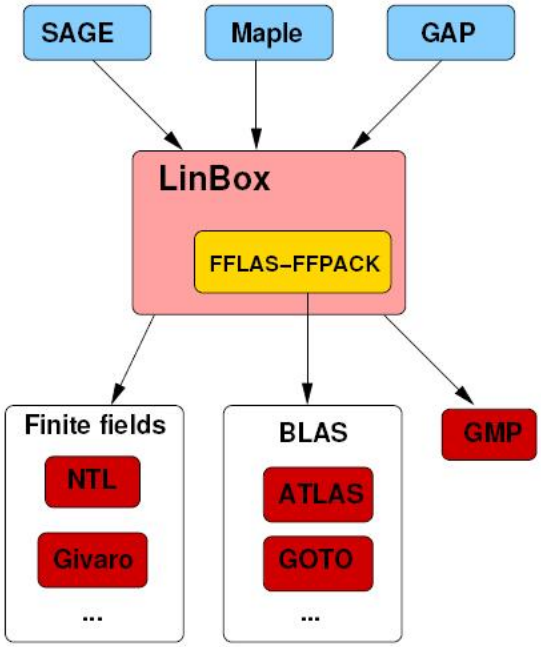

Figure 3. LinBox software architecture diagram.

1.2 The Linbox Library

People

Clement Pernet (speaker), David Saunders, Erich Kaltofen, Jean-Guillaume Dumas (speaker in algorithms session).

Project

LinBox is a library for exact linear algebra. LinBox is a joint project between the U.S.A., Canada and France, under NSF and other funding. It was started in 1998. There are about 5 active software developers. LinBox consists of around 125,000 lines of C++ code.

LinBox is available online at http://linalg.org. [12] LinBox licensing is LGPL. There are LinBox-related Google groups: linbox-use and linbox-devel.

Topology Capabilities and Applications

LinBox can compute Smith normal form and echelon form, which provide Betti numbers. These are Monte Carlo computations.

LinBox provides exact linear algebra operations such as matrix multiply, smith normal form, and charpoly over $\mathbb{Z}, \mathbb{Q}, \mathbb{Z}_p$, and GF($p^k$). Here GF is the Galois field with p^k elements. It works with dense, sparse, and black box matrices.

There is a growing application demand for LinBox capabilities including those working with simplicial complexes, number theory, crypto, graph theory.

Multiple usage models are supported: web-server, stand-alone executable, and library interface. The user interfaces are functions of matrices.

LinBox is a middleware library. LinBox lives in the space between specialized libraries like BLAS, GMP, and Givaro and top-level software such as Matlab, Maxima, SAGE, Debian, and GAP. SAGE is an open source 'end user' tool that uses LinBox beneath. In turn LinBox uses libraries for operations over rings.

Design

LinBox is designed around algorithms (e.g. rank, determinant, minpoly), domains of computation ($Z_p, F_q, \mathbb{Z}$) and matrices (dense, sparse, black box). It uses templates. Domains are templatized on element types. Matrices are templatized on domains. Algorithms are templatized on matrices. There is a field and ring (e.g. integers) plug-and-play interface for Givaro, NTL and Lidia. There is also a generic BLAS interface for floating point. An automatic process picks the algorithm based off of the type of matrix and domain based off of type of solution and component implementation. There are several levels of use: one can call executables from an end-user shell, or within code one can call a solution or a specific algorithm.

Black box or matrix-free methods

Black box matrices are linear operators, providing fast matrix-vector multiplication on the right or left, but not access to matrix entries. Hence iterative methods are required as solvers, and elimination methods are incompatible.

For the minimal polynomial, the Wiedemann Algorithm [22] is used. Other functions are built on top of this in the natural way. The Block Wiedemann Algorithm resembles the block Lanczos iterative algorithm. It is considered slow and sure. The slowness is because the number of steps is the degree of the minimal polynomial. However, without round off error, several options become available. It is safe to compute minimal or characteristic polynomials.

With structured matrices there is a fast apply $E(n) = O(n \log n)$ and sparse matrices have fast apply and no fill in. However, for black box matrices there is no access to matrix entry coefficients M_{ij}, trace, elimination, and matrix-vector multiplication is black box composition. In particular, black box matrices do not support matrix-matrix multiplication such as building up an explicit basis for a homology group representatives. Building blocks include minimal polynomial [Wiedemann 86], rank, determinant, and smith normal form. There is also a transpose operator.

For a description of the smith normal form algorithms, see "Computing Simplicial Homology Based on Efficient Smith Normal Form Algorithms" [6] by the speakers team members.

LinBox also provides echelon form, which is faster than computing full smith-normal form. Echelon form is also described in Carlsson-Zomorodian [23].

Dense matrices

When working with dense matrices the building blocks include matrix multiply. For computations over finite fields, there are several optimizations: delayed modular reduction, floating point arithmetic, and cache tuning. A finite field element fits in one machine word, which provides certain memory and computational efficiencies. One can also use "unstable" floating point algorithms, namely the sub-cubic algorithm Winograd, to get faster performance than pure BLAS. The speed over finite fields is comparable to the computation speeds over floats.

Lifting techniques, such as multi-modular reconstruction and p-adic lifting make it possible to do computations over other fields such as the rational numbers, $\mathbb{Q}$. For multi-modular reconstruction over scalars and vectors, either early termination with a user-specified probability of success or a deterministic method can be used. Lifting over the integers is needed for getting actual generators, not just ranks.

Sparse matrices

There are two approaches to sparse matrices: black box with no fill in; and sparse elimination with local pivoting strategies and the ability to switch to a dense representation when needed.

Plans

Block Krylov projections are part of the evolution of LinBox and are a building block of most recent algorithm advances and provide better balance of efficiency between black box approaches.

GF(2): The M4RI (pronounced as "Mary") library has a packed representation of elements, greasing techniques, cache friendliness, SSE2 support, and sub-cubic matrix arithmetic.
GF(3,5,7): there are similar projects.

GF(p), $p < 2^8$: Kronecker substitution.

LinBox 2.0 will include new algorithms; clean up simplify existing code; unify usage of block-Krylov/Wiedemann; redesign dense matrices to enable packing; and provide support for new architectures such as GPU, GPU + CPU, and multicore. One goal is to add packed matrices over small finite fields - this requires a redesign of dense matrices!

Group Discussion

Scalability: sparse matrices work with close to 1 million entries.

"Rank" in LinBox is a Monte Carlo algorithm, correct with some high probability depending on how much sampling computation one can wait for. It is an open problem to give a certificate of rank.

The workshop participants were asked who had used LinBox before, and it was surprising that Shawn Martin (for his cyclo-octane talk) was the only one outside the LinBox team who came forward. LinBox was robust and scaled well for his application. Perhaps now that the workshop participants are more aware of LinBox this will change. The idea that the community should share software (and use it) came up again during the panel discussion.

Q. What sort of topology objects (e.g. simplex generation from point sets) are available?
A. None, only linear algebra tools are provided.

For Smith normal form, LinBox does not compute the basis. That is, LinBox can provide Betti numbers but not generators. In the worst case it is dense and takes cubic time to compute. Having this capability would be very helpful. This involves rational numbers, and would be implemented using p-adic lifting. Or it could involve a huge transformation matrix. An alternative approach could involve cohomology, which is much more numerical.

It is hard to compare persistence and LinBox algorithms, both their capabilities and complexities.

Additional Opportunities

David Saunders is talking to Heidi Thornquist, David Day, and Michael Heroux (maherou@sandia.gov, Trilinos project leader) about a LinBox-Trilinos collaboration and summer student, combining LinBox's expertise in finite fields with Trilinos parallel algorithms experience.

Figure 4. Trilinos logo.

1.3 Trilinos

People

Heidi Thornquist (speaker), David Day, 33 others. See the Trilinos website [9].

Project

Trilinos is a large numerical data and algorithms framework from Sandia National Labs. There are about 35 developers. Trilinos was organized as a framework 10 years ago, but some solvers in it are older. It started as a solver library for PDE-based simulations, and has grown rapidly in scope. There are 50 packages now, covering preconditioners, eigensolvers, non-linear solvers, load-balancing, matrix and data types, etc.

The focus has been parallel computations over real fields (floating point computations). Trilinos supports sparse matrix, dense vector linear algebra.

Trilinos is available under LGPL licensing. It is useable as a stand-alone or a library. The learning curve is significant. Packages are atomic units. Package categories include discretizations, methods, core and solvers. The 50 Trilinos packages have a variety of inter-dependencies, but the intent is for solution developers to have to only incorporate the subset of the library that is essential for their application.

Templates are used to support multiple data types; various floating types have been tested but no discrete types to date. About 30% of Trilinos has been converted to use templates. Templates can cause compile time to be expensive; however there is no run-time penalty. In contrast LinBox uses a “traits” mechanism; traits may be more general but more difficult to read. LinBox represents field elements with objects, e.g. LinBox’s “F.muliply(g)” vs. Tilinos’s “F * g.”

Teuchos Package

Teuchos is a tools package on which most Trilinos packages rely. Teuchos includes Lapack and blas wrappers, the Parameterlist class, FLOP counters and timers, dense matrix and vector classes, smart pointers, and ordinal, scalar traits support (definition of zero, one, etc). More specifically any data type that defines zero, one, addition, subtraction and multiply, can use nearly all blas functions.

This is important because this is where arbitrary data types are supported.

There is a communicator interface, providing an abstract interface to MPI. Trilinos does not have an MP_INT interface. Trilinos does have an interface to GMP.

Arprec has been used through this interface. Arprec uses array of 64 bit floating point numbers and GMP provides integer rational and floating point numbers. The Comm class is a pure virtual class that handles gather/scatter however implementation is provided separately in subclasses.

Petra Packages

Petra provides a common language for distributed linear algebra objects and a common interface for linear algebra services. Variants include the following: epetra is essential petra, tpetra is templated petra, and jpetra is java petra. Tpetra provides abstract interfaces. It contains comm objects and maps describing layout. However the packages must be implemented using abstract interfaces. Tpetra will be released fall 2009.

Kokkos Package

The Kokkos Node package does the linear algebra computations requested through tpetra on local compute nodes. Kokkos is behind the interface, hidden from users.

Zoltan Package

Zoltan provides (dynamic) load balancing, graph partitioning, data migration, and matrix ordering.

Belos and Anasazi Packages

Belos is a templated linear solver library. Everything in Belos is a traits class. Anasazi is the similar library of block Krylov algorithms for eigenvalue problems. The interface requires matrix vector multiplication by the matrix and its transpose.

Trilinos for Topology?

In terms of using Trilinos to address computational topology work, much effort has been spent on the real field. If Tpetra were to operate over the rationals, integers, $\mathbb{Z}_2$, or $\{-1,0,1\}$), then new kokkos nodes would be required for local computation.

GMP is already integrated into Trilinos, but using GMP may be problematic for parallelism. There is another issue with GMP because of its dynamic data types: a scalar might consume a very large amount of local memory.

Group Discussion

The LinBox memory model is unique.

Some Trilinos design decisions were limited to different kinds of floats.

Q. What are the possibilities for integrating LinBox and Trilinos?
A. The first thing is to write wrappers to access LinBox from Trilinos.

1.4 Reeb Graphs

People

Attila Gyulassy (speaker), Valerio Pascucci (speaker in Algorithms section). The broader Morse theory and combinatorial topology efforts involve perhaps a dozen people, including Peer-Timo Bremer(LLNL), Ajith Mascarenhas (SNL), Gunther Weber(LBL), Janine Bennett(SNL), and a collaboration with Scott Dillard (LANL).

Jackie Chen, David Thompson, Ray Grout, and Ajith Mascarenhas at Sandia National Labs are using these techniques for studying flame/combustion turbulence, ignition, and extinguishment, as described in their Applications session talk; see section 2.1.

Project

These techniques are used to analyze a topological domain together with a scalar field over the domain. They are useful for identifying the interesting feature space of a function. These techniques are based on the topology of level sets, and the gradient field.

The techniques have been developed by the computer graphics communities. The software in this talk produces astounding pictures that are more directly user-interpretable than, say, Betti bar codes from persistent homology.

Methods and Results

The general pipeline used when working with these algorithms consists of 4 steps: data acquisition, compute topological representation, filtering/simplification, and final analysis. The Reeb graph could be thought of as a dimension reduction technique.

Given a scalar field at a particular time step, a Reeb graph reduces the dimensionality of high-dimensional data by providing a structural view of the connected components of level sets and by providing insight into the number of connected components. When the domain is simply connected, then the Reeb graph is called a contour tree. There are other variants of Reeb graphs such as merge and split trees. These specialized versions receive attention because they can be computed more easily, and lend themselves to efficient streaming computations. The most general case of Reeb graphs is not solved.

The Morse-Smale complex is a different segmentation, consisting of a sort of dual cells to the Reeb graphs: it groups regions of the domain that flow to the same critical points, describing the structure of the gradient field. MS complexes allow for exploration of ridge lines and basins. The underlying theory is Foreman's discrete Morse theory. Building a Morse function sometimes requires changes to the data for them to meet the necessary conditions. For example, simulation of simplicity is often used to ensure critical sets are points.

The resulting structures contain all features, often too many uninteresting features and noise. So filtering and persistence measures are used to identify the subset of features of interest.

Language issues: Persistence and Filtering

The word "persistence" is used in both persistent homology calculations and in Reeb graph simplification. Both meanings are related to the ideas of a topological feature's significance, longevity, and sensitivity to small changes in the input. In homology, persistence describes the creation and destruction of homology groups during the steps of the filtration. In Reeb graphs, persistence describes whether a critical point would still be critical if the scalar field over the dataset were perturbed.

A filtration in homology is an ordering of complexes by inclusion.

Filtering in Reeb graphs is a selection of a subset of the topological features. E.g. one can search for links in the Morse-Smale complex of specific types.

Simplification means the cancellation of a pair of critical points, usually based on both their scalar field values and their proximity in the domain. Simplification may also be directly prescribed by a user of the software. Simplification of the scalar function and/or the resulting topological structure can be performed by decreasing the number of critical points via an elimination of a pair of critical points. This elimination can be persistence-based or can be based on more complicated measures (e.g. hypervolumes). Simplification may have non-obvious consequences to the actual topology of the graph. Note that simplification of the structure may not require modification of

the original scalar function as many people want an idea of the structure and don't care to actually smooth the function itself.

Applications

Tools from Morse theory provide a structural view of scalar functions and are applicable to a broad scope of applications including fluid flow through porous media, tracking flame surfaces, turbulent mixing, tracking combustion (2D), surface re-meshing, and CAD verification. The tools provide a roadmap for exploring data; however they still require an expert user behind the wheel to explore feature space.

There are many specialized standalone applications, targeted to particular application domains. A core algorithms is the streaming computation of Reeb graphs; the talk described a solution that is the state of the art for 2D and the best known (perhaps only) solution for 4D+. It only works for simplicial meshes. The scalability of the algorithms depends on the order in which the algorithm processes the mesh. A 3D version is being built which is optimized for manifold tetrahedral meshes without boundary. Volumetric meshes of any type are supported. There is also software for Jacobi sets; these sets are a way to deal with multiple scalar fields simultaneously. Another algorithm computes the merge/split tree in 3D+time using a streaming construction. A threshold is used to simplify and track overlap in segmentation. The algorithm works over meshes of general type, and in theory in arbitrary dimensions.

The streaming computation of Reeb graphs is discussed by Valerio in his Algorithms talk.

Further Opportunities

The group that is developing these applications and codes is very diverse. There are lots of code for the different types of trees. Many of these were developed by different graduate students, and were separate stand-alone efforts. The software is not currently available to the broader community.

Jason Shepherd is coordinating an effort by Valerio Pascucci, Claudio Silva (Also Univ. Utah), and their students / post-docs to put much of the software described in this talk into VTK / Titan [20, 11]. This will make the software available to the broader community. Also having the software organized in a new modular way will be useful within their projects for developing new applications. Currently there are dozens of packages, with a relatively narrow usage range for each. Currently these packages do not interact with one another. For example, a useful package would be a re-usable module that created a 2d Reeb graph for streamed input data.

Janine Bennett, David Thompson, Philip Pebay (Sandia National Labs) and Maurice Rojas (Texas A&M University) and Valerio Pascucci have a new project to construct Reeb graphs over locally-averaged (statistical, non-scalar) quantities.

1.5 Geomagic

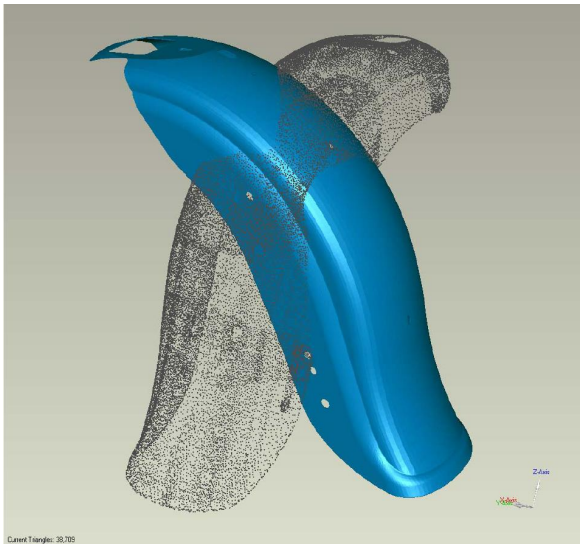

Figure 5. Alignment of congruent shapes in Geomagic.

People

Yates Fletcher (speaker). Plus others at Geomagic [8] including Herbert Edelsbrunner.

Project

Geomagic is commercial software. Yates was Geomagic's first North Carolina employee, circa 1999.

The talk was titled "A Mathematical Tour of Geomagic Software." Geomagic is software for solid models, digital shape sampling and processing. The theme of this work is the transformation of geometry to linear algebra.

One capability is the generation of a solid model from point samples. Point data may come from multiple scans of the model, from multiple perspectives. Registration and cleanup is used to bring these different scans into the same coordinate system, remove outliers, prune for uniform density, and reduce noise.

Then the point cloud is triangulated. Points are triangulated by projecting each point and its neighbors to a local plane. The Delaunay star of the point in that plane is computed. A triangle that appears in all three Delaunay stars of its vertices is included in the final model. Delaunay stars also comes up in free form surface design.

Any remaining holes are filled. "Wrap" is an algorithm that is very good for models with up to 10^4 points, that do not have smooth local surfaces. A new algorithm handles up to 10^6 points. A zigzag connection can be used to fill holes. Local curvature calculations help. One can still get a non-manifold object, and the software provides capabilities to manually fix them.

A common key difficulty is reconstruction of sharp edges. One often gets overlapping or inconsistent triangles because points are sampled too densely and span the inside of the volume near a sharp edge. "Fairings" can be used to specify a model edge that follows a crease. A curvature-based Morse-Smale function is useful for model vetting, and for splitting the underlying model into different patches for different NURBs and fillets.

Another paradigm is to compare a reconstructed object to its initial point input. A sum-of-squares minimum transformation is used to produce local error plots.

The workshop participants asked a number of questions concerning the provable correctness of the techniques.

At one point several academics at the workshop pointed out that the original paper by Edelsbrunner describes a different technique. Yates pointed out a practical problem with that technique, and Geomagic relies on a patented variation that Yates developed.

Polygons are converted to NURBs. NURBS are the industry standard. A G2 continuous surface is constructed. This works by producing a network of quad patches, sampling the underlying geometry on a regular parametric grid over each patch, and constructing a NURB surface on each patch. Points are processed using local neighborhoods defined by octrees or kd-trees. Local reference frames are used for doing curvature analysis. Filtering is a selection of a subset of the topological features.

Noise reduction is performed by moving each point to its local reference NURB. Polygons are converted to CAD models to capture design intent. Numerical curvature maps, region and separator sets, contour extraction, extending contours, classification, and trimmed primaries are things that are part of the process.

For rigid alignment of congruent shapes, alignment is performed through a hierarchy of optimization problems. Congruent shapes are aligned by choosing a set of discrete samples S on the first object and using an Iterated Corresponding Point (ICP) algorithm. (ICP may be formulated in terms of quaternions.) This algorithm finds a correspondent set of samples on masters, computes rigid motion that minimizes sum over all points x in S, and applies the motion to the second object. This process is repeated until the movement is negligible. Shape warping is performed using tie points and an SVD solve is performed to figure out warp.

Computational Dental Surgery Application

An application is computational dental surgery. The problem is to warp the shape of a generic replacement tooth to fit on a crown. The tooth shape is constrained by contact with neighboring

teeth and the opposing tooth, but this does not fully constrain the problem. This is an under-constrained problem, that is resolved using least squares. Using the wrong control parameters or wrong objective function can lead to an aesthetically horrible tooth.

Research Opportunities

People at Geomagic are researching the compression of geometry in a manner similar to jpg image compression. The system under investigation uses a lattice index and offset. An adaptive distance function is built over a lattice and a shell. (The vertices in the lattice nearest to the surface are called the shell.) A bi-cubic centered (BCC) lattice, with tetrahedra, is used. Tetrahedra can be refined into similar tetrahedra, as is commonly done with octrees. Here BCC determines a quadric surface. There is a unique quadric for the four vertices and six edges. Compression keeps only cells that are shell cells and the lattice structure subdivision employed facilitates representation of quadric surfaces.

Reconstruction (uncompression) from the compressed format is a kind of interpolation problem. Reconstruction is performed by producing distance values at the finest level of shell vertices. This process uses an existing value when possible; otherwise it reproduces a value by interpolating the result from the smallest cell of the consolidated shell that contains it.

Geomagic is interested in a classification system and algorithm for primitive surfaces. Classification may be in terms of nullspaces.

Geomagic is interested in slip theory.

2 Applications

The Saturday afternoon session focused on laboratory applications, especially those that are not currently using topology, but could. Format consisted of an overview talk, together with a discussion session to explore what is possible. The intent is for application owners to engage topology experts for help in solving their problems, with the potential for longer term partnerships. Speakers were asked to address the following aspects of their applications:

- geometry, if any
- dimension: 3d, higher-d, arbitrary-d
- questions needing solution methods
- interesting structural features one would like to discover and compute

6 talks were given on Saturday afternoon.

- Morse-Smale structures for understanding large-scale 3d combustion simulations, Ray Grout (Sandia National Laboratories)
- Fracture and Fragmentation of Simplicial Finite Element Meshes Using Graphs, Alejandro Mota (Sandia National Laboratories)
- Homology of Complicated and Random Evolving Patterns, Thomas Wanner (George Mason University)
- Topology in Image Analysis for Distinguishing Tampering from Environmental Degradation of Unique Tags, Kurt Larson (Sandia National Laboratories)
- Persistent homology for parameter sensitivity in large-scale text-analysis (informatics) graphs, Daniel Dunlavy (Sandia National Laboratories)
- Discrete combinatorial optimization fitness landscapes, Jean-Paul Watson (Sandia National Laboratories)

See http://www.cs.sandia.gov/CSRI/Workshops/2009/CAT/program.html for links to some talks' slides.

2.1 Morse-Smale for Combustion

People

Ray Grout (speaker), Ajith Mascarenhas (tag-team speaker), and Jackie Chen (Sandia). Connections to David Thompson, Attila Gyulassy, Valerio Pascucci, Janine Bennett, and Gunther Weber. See also section 1.4, the Reeb Graph talk in the software session.

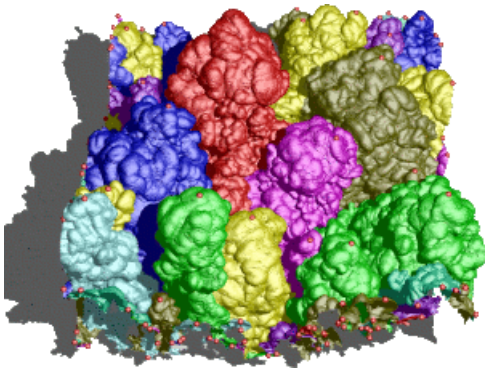

Figure 6. Turbulent mixing.

Turbulent Reacting Flows

Topological visualization methods are used on the frontiers of understanding physics. The setting is large-scale finite element calculations of turbulent reacting flows, namely combustion as an oxidizing agent and fuel mix and react, and as temperatures and jets drive turbulence and diffusion. At issue is the accurate simulation and tracking of thin O_2-fuel active interface regions over time, as reactions ignite and extinguish. The modeling of flames is important for designing efficient power plants and engines.

A challenge is that accurately modeling the small-scale reactions and the large-scale flows would require intractably large grids, 10^9 elements, over time. The devices are too big, and the physics are too small. These multiple scales must somehow be resolved or approximated. Flames have a natural length scale. Turbulence influences the flame, and vice-versa, so coupling scales is required.

Direct numerical simulation (DNS) is a learning tool for effects in simplified models, such as lifted jets. In premixed flames, the turbulent flow interacts with the flame speed. Here the goal is to design the system to keep the flow and flame going. In addition to the jets, there are also wall bounded flows.

Combustion and Topology

Topology helps one to explore fundamental questions about extinction in a flame. High dissipation is related to extinction. A contour-tree algorithm is used (Carr and Snoeyink 2001). A merge tree is determined. A medial-axis of the surface of ignition is computed. These are used to find features in the chemistry, and length scales of the morphology. In the auto-ignition of lifted jets, there is a build up of radicals in areas where there is small dissipation.

A Morse-Smale (MS) complex over 3-d is computed. Significant MS simplification is helpful

in illuminating the gross structure, in keeping with the vast scale differences in the physics.

Research Opportunities

Of the many open topics, the following five stand out. First, the size of the data sets presents challenges to segmentation algorithms. The second issue is related to vector field topology. Third, could topological tools shed light on the imprecisely defined extinction regions?

Fourth, one seeks to track related features separated in time. For example, a region ignites, experiences small-scale turbulence and also migrates downstream due to the large-scale diffusion gradient and temperature effects. Perhaps the region splits into two. Each region eventually extinguishes. Currently one is unable to track the vaguely-defined ignition region over time; the temporal lag complicates correlating one region with another. Tracking two related features over time is also an interesting open problem: where is a moving vortex with respect to a moving flame? Perhaps a combination of small time-step tracking and human-informed correlations would work.

Fifth, currently calculations are done in two passes: physics, then post-analysis using MS topology. If the physics and analysis were done simultaneously, perhaps less data (than the entire 3-d + time domain overlaid with multiple physics attributes) could be stored. Also, the analysis may inform what local scale-resolution of the physics is required.

2.2 Fracture of Meshes

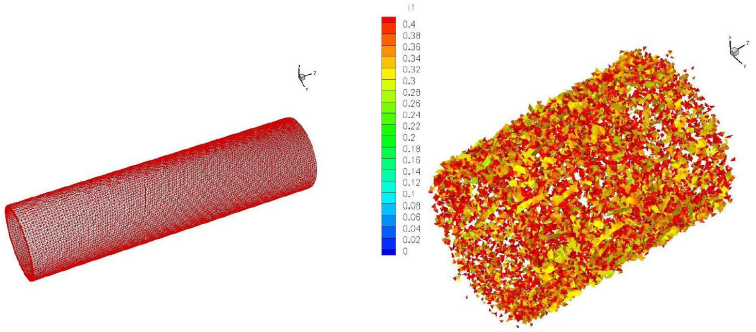

Figure 7. A fractured mesh from an exploding tube.

People

Alejandro Mota (speaker). Also Jaroslaw Knap (Sandia National Laboratories). Sandia has an established long-term effort in modeling fracture mechanics.

Graph Fracture Algorithm

The title of the talk was "Fracture and fragmentation of simplicial finite element meshes using graphs." The setting involves the finite element simulations of solid materials undergoing stress or blast-induced fracture and fragmentation. Alejandro developed a technique for modeling material fracture over time by splitting the mesh along approximate material fault lines. For example, a crack initiates by splitting a mesh vertex into two vertices, and splitting a subset of the star of higher-dimensional elements containing that vertex into two, each containing the original vertex or its split twin as a subface. More splitting of faces of various dimensions may occurs subsequently. The key is to ensure that the mesh remains a well-defined simplicial complex, with non-interpenetrating geometry, at all times. The main focus of the talk was a description of how to split a simplicial mesh consistently under all the cases of local events that can occur.

Note that material fracture is modeled only along element interfaces, not within an element. The implementation involves higher-order tetrahedral elements. The implementation uses a 4-way linked list. (See Pandolfi and Ortiz [16]).

Simplicial meshes are graphs of adjacencies. Simplicial complexes arise naturally. Signed incidence matrices represent the finite element mesh. Ordering, even local ordering, is more important than with the traditional nodal element representation. The graph is directed in order to account for orientation (e.g. the sign of the result of the boundary operator).

Non-manifold topologies have always been a challenge; this is addressed by the graph approach.

There is a parallel algorithm. It is efficient and consistent with the serial algorithm. This is the only parallel fracture algorithm of this type.

Research Opportunities

Is there a way to do this that also works for material contact, and that is scalable?

The Boost graph library is used for handling graph primitives. There is a parallel Boost graph library. But it appears that this is currently not being used due to inadequacies of parallel Boost, but Alejandro had to do a lot of work to do similar things. Follow-up discussions between Alejandro and the Titan [11] team (Jason Shepherd) on this topic are planned.

It would be desirable to extend the algorithm to CW complexes and to hexahedral meshes. Hexahedral meshes are desired for structural mechanics because of their computational efficiency (fewer and lower-order elements) for a given accuracy requirement.

Another problem of interest that came up in the group discussion is the identification and tracking of topological features as a material fractures. I.e. track the topology of voids as they form and coalesce. This may be more interesting for fracture (cracks in a solid) than fragmentation (many isolated pieces of a solid). This may provide more accurate estimates of material strength

than the current practice of estimating the local volume fractions of material and void.

2.3 Homology of Evolving Patterns

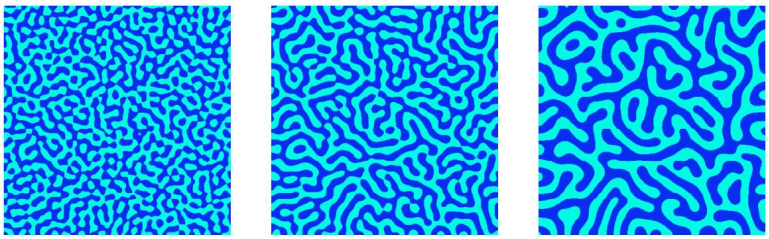

Figure 8. Homologically-assessed time-evolving microstructure patterns.

People

Thomas Wanner (speaker). CHomP team [13]. See the last talk slide, "Thank You." See also Konstantin Mischaikow's talk in the Algorithm session, section 3.5

Homology

The title of the talk was "Homology of Complicated and Random Evolving Patterns." Homology, specifically Betti numbers β_0, β_1, and sometimes β_2, are used to characterize patterns in material features. Cubical meshes (pixels) in two and three dimensions are used. The main algorithm involves co-reduction. Cubical meshes allows for very efficient co-reduction before homology is calculated. Relatively large Betti numbers are encountered: e.g. $\beta_0 = 1, \beta_1 = 1700, \beta_2 = 0$, for 10^5 elements.

The authors tried Plex before, but ended up having a student implement the Zomorodian-Carlsson [23] algorithm specifically for this setting instead. The main issue was getting a filtration for the cubes in this context. Thomas's general comment was that as the complexity (number of elements? Betti numbers?) increased, the performance tradeoffs between Zomorodian-Carlsson and co-reduction switched, so both algorithms have their contributions to make.

Applications and Validation

Complicated irregular patterns can be observed throughout the applied sciences, for example material science. Fluids experiments and simulations exhibit spirals and defect chaos.

The talk covered several specific applications. The question of the validity of these simulations arises in each problem. The general validation technique is to compare features of the simulation with features of experiments. To generate these features, a common paradigm is to threshold a quantity to binary $\{\text{white}, \text{black}\}$ values, then study the patterns of regions with each value. A commonly used measure is the Euler characteristic. However, Thomas shows that this is inadequate because it doesn't, and can't, distinguish between the boundary and interior effects. In contrast, Betti numbers can.

The time-evolution of Betti numbers is studied. A time series is generated that demonstrates the average evolution curves for the number of internal components and the number of components touching the boundary. The time series of Betti numbers for experiments and simulations can be directly compared, and shows that certain simulation methods are more accurate. For example, in the phase separation model, simulations using a stochastic model can be shown to be more accurate than ones using just a deterministic model.

For example, in some settings Betti numbers are expected to increase or decrease depending on the particular dynamics and initial and boundary conditions. These measures also appear to be consistent with human visual evaluation of the similarity of patterns produced by different methods and experiments.

Phase Separation Models

Consider phase separation in a binary alloy. Can topology give new measurements for phase transitions in materials? Yes, using the first two Betti numbers.

Evolving microstructures models are used to study phase separation. Quenching of homogeneous binary or multi-component alloys may lead to phase separation, generating complicated microstructures. The resulting patterns are generally transient. A variety of phenomenological models for such processes have been proposed over the years. The goal is to perform some kind of model assessment and to make quantitative statements as to whether or not the models are valid.

The two dimensional Cahn and Hilliard (1958) model with a stochastic source term is studied using CHomP, and compared to a deterministic model.

Homological analysis of evolving microstructures was performed. Each pixel's continuous variable is thresholded to one of the two states to give a discrete structure for display and on which to perform homology calculations. Even for relatively small 3D microstructures, the Betti numbers are large and non-obvious, and have to be determined computationally.

Averaged Betti numbers show decay, representative of parabolic partial differential equations.

Questions about the boundary arise naturally. It is possible to distinguish the homology due to the boundary, the boundary components, from components due to the interior. The averaged Euler characteristic is entirely due to boundary effects for this particular problem, and so are insufficient. Betti numbers are necessary.

Cahn Hillard has a singular perturbation parameter. Epsilon scaling studies pick up on small β_i variations. If the perturbation parameter is, in a sense, scaled out of the problem, the topological methods are even more powerful.

Poly-crystals

The microstructure of calcite crystals is that they expand in one direction, and contract in another direction. This causes failure of marble structures, such as a building wall exposed to daily heating by the sun. Thermal degradation of marble is caused by internal stresses in polycrystalline materials that can lead to micro cracking and ultimately to destruction of components.

Beta-eucryptite composites are expected to have a zero coefficient of thermal expansion. However, we see spontaneous material ejection, resembling a dry volcano, after an indentation due to internal stresses. Hence being beta-eucryptite is shown to be insufficient to explain observed behavior.

It is the orientation of the grains that distinguishes these physics, not the atomic-level microstructure. Even identical grain microstructures can lead to considerably different elastic energy density/stress networks and therefore to different cracking behavior. Internal stress networks, formed over time, due to the thermal and load history, control the physics.

A homological analysis of the grain boundary orientations confirms this. Grains are chosen randomly and assigned an orientation - internal stress networks show different stress. The length scales on internal stress networks has impact on failures. Grain boundary misorientations are studied. The effects of domain size are controlled through rescaling.

Rayleigh-Bernard Convection

Spiral defect chaos in fluids is studied. The same approach of thresholding patterns and computing Betti numbers is used as in the prior applications. The connected components and loops for the cold down-flow and hot up-flow is visualized. Experimental time series between up-flow and down-flow exhibit a surprising asymmetry; however typical simulations preserve symmetries - evidence that the simulations are not sufficiently valid for this phenomena. The Betti number time series indicates the breakdown of the Boussinesq approximation.

Here smoothing is used to eliminate small generators. It would be interesting to evaluate persistent homology methods for this problem.

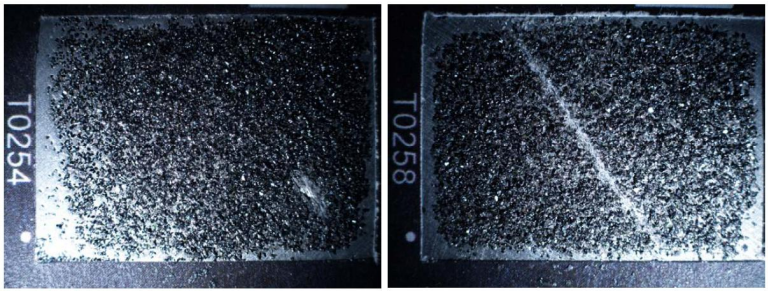

Figure 9. Reflective Particle Tags (RPT).

2.4 Image Analysis of Reflective Particle Tags

People

Kurt Larson. The Sandia effort on producing and analyzing these tag involves many people and spans two decades. See the nine "coauthors" on the first talk slide.

Summary

The idea is that topology might be useful for the analysis of images of certain reflective particle tags (RPT). These tags were developed in the 1980s at Sandia National Labs. The unique-identification of these tags is a solved problem. Distinguishing between a tampered, damaged, and unblemished tag is an open problem. This application is not yet using topology. It is not yet clear that Algebraic Topology solves these open problems. The title of the talk was "Topology in Image Analysis for Distinguishing Tampering from Environmental Degradation of Unique Tags."

Setting and Tag Definition

The setting is Nuclear Nonproliferation applications. Critical materials are often distributed for legitimate purposes. In addition to materials, sometimes sensor systems are sealed: attribute measurement systems are used in disarmament and it is vital to know that the systems have not been tampered with. Historically, welds on containers were used, but it has become clear that welds may be tampered with in ways that are impossible to detect. Instead these objects are encased and sealed with tags.

Specific portions of the encasement are "sealed" with a reflective particle tag (RPT). This is a spray-on adhesive resin with thousands of small optically-significant crystals embedded. The pattern of crystals make tags unique from one another; there are a large number of grains, and their orientations are unpredictable. An individual tag cannot be duplicated. The crystals are chemically

unique (natural source), so neither can a new "tag" be produced by a third party that appears to be from the family of legitimate tags. The resin becomes very brittle when cured; tampering with the tag will irreversibly damage it. Tags look similar to 400-grain sand paper.

There is high confidence that tags are identifiable, unique, irreproducible and can't be moved or tampered with without evidence.

The hardware used to install and image tags in the field must be (and currently is) small enough to be carried in and out by the team applying the tag.

Identifying tags is currently done using tools from signal processing and machine learning. Controlled imagery using a single position (tags are augmented with reference points for positioning a camera) and a variety of illumination patterns provides complicated patterns of reflections and bright and dark spots. Some of these reflections appear and go extinct in just a few degrees of rotations. When two images are of the same tag, there are about 10^4 matches in the image space. When the images are of different tags, there are about 10 matches. Hence tags are highly distinguishable.

Is there room for topology to help out?

Identification, Damage, and Tampering

How do you verify whether unattended seals (reflective particle tags) have been tampered with? What if the host organization of the sealed containers is uncooperative? Tags may be in harsh environments that degrade them, such as being exposed to industrial processes, or sitting outside for years in a dusty wind. The curious may put fingerprints on tags. The careless may scratch tags. How does the inspector have confidence in the integrity of a tag? How is inadvertent damage distinguished from malicious tampering?

False confidence is worse than no confidence. Both false negatives and false positives could have serious political consequences. However, automated technology could be used as a filter: automated tools could indicate when additional (perhaps human) inspection is required.

Identification of tags is easy; however it is difficult to distinguish between inadvertent damage and malicious tampering. What are signatures between malicious and inadvertent changes? Can one have confidence in the integrity of the tag patch? What does it mean to say that a tag has integrity or has lost it? What is the relationship between match quality and confidence?

Afra suggested looking at distances in metric space and suggested taking local data measures in order to infer global meaning.

The RPT could be decomposed into overlapping patches.

Michael Robinson suggested using cohomology classes to identify cuts across a patch. For example, say you have 37 cut paths from top to bottom. Can we then say that there is no apparent damage in that direction? As in the game of "hex," an unbroken path from one side to the other of

undamaged material implies that there does not exist an unbroken path of damaged material that crosses it.

Again, going from local to global information implies topology might be helpful.

Would it be possible to assign a depth to parts of a tag? For example, the top surface of a tag might be sand-papered, leaving only the deep features. In that case could one still be confident in the tag?

It was noted that algorithms should account for as much of the detail in the structure as is possible so that adversaries cannot learn ways to tamper with the tag undetected.

Valerio asked about the availability of a library of tampered patch images to gain insight into what structures might appear in the patches when tampered with. The answer was that some images can be shared (some are in the talk slides), but an image library will never span the space of possible changes. Indeed a characterization of the space of possible ways humans might attempt to tamper with the tags is probably unknowable, despite having high confidence that if tags are tampered with then there will be evidence.

Open questions

1. Due to IAEA (International Atomic Energy Agency, http://www.iaea.org/) operating principles, much of this information is and will be in the public domain. It would be helpful for algorithms and image hardware specifications to be public, in the same sense that public cryptography and open software is more trusted. Having an open process also motivates the use of all of the available complexity in the images, so the process can not be broken by simple tricks.

2. What is the relationship between RPT match quality and confidence?

3. What does it mean to say that a tag has integrity, or that it has lost it?

4. When is a tag degraded, damaged, unreliable or tampered?

5. If we were to add micro-machined materials to the tags to augment their imagery signature, for example to more easily distinguish between tampering and damage, what properties should the materials have?

Research Opportunities

Workshop participant Carl Diegert has acquired some tag images across view angles. Michael Robinson has started follow-up discussions and activities with Carl and Kurt Larson. Afra Zomorodian had some suggestions for investigation as well.

2.5 Persistent Homology of Text-Analysis Graphs

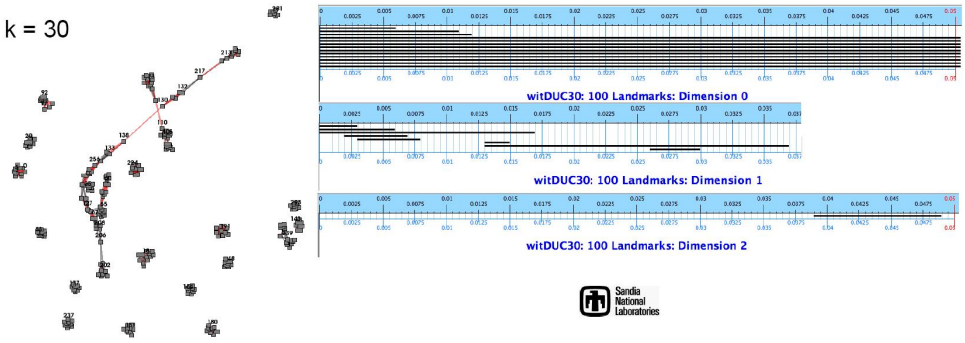

Figure 10. Left: A text-analysis graph. Right: the JPlex witness stream Betti bar graphs.

People

Daniel "Danny" Dunlavy (speaker). David Day made some of the initial explorations of text information graph persistent homology using JPlex. Also Scott Mitchell.

Summary

The title of the talk was "Persistent homology for parameter sensitivity in large-scale text-analysis (informatics) graphs." The idea is that topology might be useful for the analysis of a corpus of text documents. Algorithms produce a graph of the (information in) the documents. These algorithms have parameters. Analysts would be mislead if the displayed graph had a structure that was the result of a peculiar choice of parameter values, so analysts seek to select values in the middle of a stable and "representative" range. Currently this is done manually.

Initial experiments indicate that perhaps persistent homology can help automate parameter selection, and produce other insight. A key challenge is what complex and filtration to use.

Text analysis is a common informatics application at Sandia National Labs. This application is not yet using topology. It is not yet clear that topology will help.

Text Analysis

Text analysts start with a corpus of text files and seek to extract knowledge from it to answer a particular question. Analysts prefer to read files to avoid any bias introduced by tools. A file may

be subdivided into several documents. Analysts manually interpret individual documents. This traditional approach doesn't scale, as there are millions of documents and analysts are busy, so automated tools are used to select which documents analysts will focus on.

The data to be analyzed is very large scale; a small data set will contain 700k documents with 600k features. Any reasonable algorithms should scale to this size and larger.

The text-analysis pipeline consists of six steps: ingestion, preprocessing, transformation, analysis, post-processing, and archiving. This work focuses on the first three stages.

Data clustering, classification, and summarization are useful. (This is different than PCA in that the goal is to capture the information presented in a document using a small amount of work.) Sandia uses some tools and techniques for hypothesis testing, visual analysis, surrogate data generation, and model verification; these are active research areas. It is important that algorithms provide a mechanism to incorporate analyst knowledge through annotation and relevance feedback, metric learning and priors.

Latent Semantic Analysis (LSA) is a central tool. It is similar to PCA. It will produce a relational graph from text documents. It is used to discover document-to-document, term-to-term, and term-to-document relationships in the data. Some analysis alternatives to SVD include probabilistic modeling, multi-way modeling, semantic graphs and parafac tensor decomposition. These give directed edges and multiple meanings on edges. (How would these affect homology computations?)

Prior to LSA and other techniques, one must pick the terms. This involves making many discrete decisions. Many choices are based on linguistics, but many are ad hoc. Common techniques involve natural language processing such as named entity extraction, sentence boundary detection, stemming, lemmatization, and part-of-speech tagging. Furthermore, there are options to use n-grams or words, keep or eliminate numbers, and to treat paragraphs or fixed size chunks as "documents." Feature weighting is a way to avoid some types of bias, e.g. for the size of a document. Natural language processing is a source of instability. The filtering of the text is not a well defined problem. Data imperfections such as encoding, segmentation, and incomplete data pose additional challenges.

LSA uses a vector space model that starts with a rectangular term-document matrix A. Each element a_{ij} is a measure of the "importance" of term i in document j, such as the number of times the term appears in the document. The matrix A is factored. The truncated LSA score is obtained from the truncated SVD. This decomposes the information space into concept vectors (a.k.a. singular vectors). One algorithm choice is how many concept vectors to use, the rank of the decomposition. Past work on selecting the right rank is designed to capture the variance. This might lose or bury the structure we are looking for. However, methods that clearly reveal specific structures will omit many significant relationships.

Document similarity graphs are constructed by treating each document or term as a vertex. A complete weighted graph is produced. The weight of an edge is a similarity measure between the documents (terms), dependent on linear algebra operations using the concept vectors.

These weighted edges are filtered, usually by weight and by ensuring each node has a minimum number of neighbors, to produce an unweighted graph. For large problems, thresholding is performed on the fly for each document because the entire graph cannot be stored.

The graph is explored using visual analytics. LSAView software is used to look at the graphs and associated adjacency matrices. The graph is laid out in 2d and the colors of edges between documents indicate the similarity measure between the documents. As the user tweaks parameters and re-runs the methods, a goal is to identify the most and least significant changes. An example of a structure that may change under perturbation is two heavily linked clusters connected by a single inter-cluster edge. Once this edge goes away how does this affect the structure of the models?

A problem with this approach is that modifying control paths for modeling parameters can generate dramatically different views of the same data. There are many choices, both continuous and discrete, that impact the results. One goal in looking into topological methods is to discover relational graphs that are stable in the presence of the perturbations of these parameters.

An added challenge is that there is no "correct" view of the data. Different views may be preferred depending on personal preference and the question the analyst is trying to answer.

Analysts seek an understanding of the sensitivities of these tools to all these parameters and issues. Ideally, analysts would like to know how sensitive their interpretations of the data are to these parameters. Can persistent homology provide insight?

Initial Results

A small text corpus was analyzed using the above approach. Using the weights of the complete graph from LSA, a Witness complex filtration was constructed. Persistent homology was calculated using JPlex. Some Betti barcodes were shown. It was possible to see some stable and unstable (noisy) regions in parameter space. Sampling the complex appears to lead to different results.

Application Specific Research Opportunities

We are interested in discovering the relationships between the data. It is hard to find features without expert knowledge. There may be dire consequences for false negatives.

Is there useful topological structure in these graphs? Can we associate topology with the types of things that analyst want to discover? Do different views cause people to make different conclusions?

How can an analyst add her own knowledge into a data set, e.g. how do you incorporate user modifications into the document-term matrix? Can the knowledge about the structure of one data set be used to inform the analysis of another related data set?

If a filtration is built, does the order induce a bias?

This problem is hard, and all the reasons why are not well understood.

Topology Research Opportunities

What are the most meaningful topological features for text analysts, how would we compute them, and how would we display them? It is unclear what topological structure means in the context of text analysis. E.g. suppose β_2 is 2, what should text analysis practitioners think about their data, and what should they do to further analyze it? What about generators? Can we compute generators that provide application-specific insight? Text analysts have some familiarity with how to interpret clustering and shortest paths and perhaps connection subgraphs.

What challenges do the scale of the data pose for persistent homology? The fact that the weighted text similarity graph may be complete is problematic. Producing a witness complex on a filtered complete graph is even more problematic. Perhaps Afra Zomorodian's ideas of smaller alternatives to the Witness complex would be useful. Since the weighted graph is filtered on the fly because it is not possible to store it explicitly, it might be useful to develop streaming versions of persistent homology.

Scott Mitchell, Janine Bennett, Eric Boman, David Day, Tamal Dey, Daniel Dunlavy, Robert Ghrist, Shawn Martin, and Valerio Pascucci wrote a proposal to research some aspects of the topology of informatics graphs.

Some of the paramaters, such as the weight-threshold for including edges, lend themselves well to persistence because as the parameter is increased, the complex gains more edges. Other parameters, such as the number of concept vectors to use in the truncated SVD of LSA, have a more complicated (non-monotonic) effect on the complex. Can zigzag persistence [2] or other variations of persistence be used to explore these parameters?

Jenifer Kloke is interested in using zigzag complexes, Stanford's DataMapper and topology tools to explore some of these graphs.

2.6 Topology of Cyclo-Octane Conformation Space

People

Shawn Martin (speaker), Evangelos "Vageli" Coutsias, and Jean-Paul Watson (Sandia National Labs). Also William Mike Brown (Sandia) and S. Pollock (UNM).

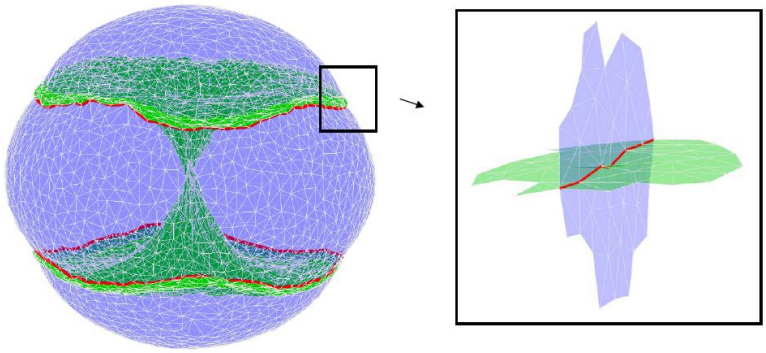

Figure 11. Left: A 3-d IsoMap [1] view of a triangulation of the minimum-energy cyclo-octane conformation variety. Right: a remapping of a neighborhood about a self-intersection.

Summary

The title of the talk was "Topological exploration of the variety of minimum-energy cyclo-octane molecular conformations." Conformations are the shapes that a molecule can take. The main idea is that topology might reveal the structure of the space of possible shapes. The work might be characterized as algebraic geometry and dimensional reduction, with some topology.

This test problem did use some topology. Topology helped, especially since visualization and other tools had difficulty with the high-dimensional nature of the problem.

Cyclo-octane

Cyclo-octane is a small and well-studied molecule. It is an eight member ring C_8H_{16}. It has three stable groups of conformations, named after their shape: boat, boat-chair, and crown. One can enumerate these conformations analytically by assuming bond lengths and angles are fixed while bond rotations are varied. One can write the algebraic equations that lead to the variety defining the conformation space. The conformation space is thought to be two-dimensional due to a ring closure constraint, by counting degrees of freedom and constraints. Solving these types of equations directly is not generally possible currently.

Methodology

Vigeli generated millions of individual conformations (i.e. point solutions to the variety) of cyclo-octane. These oversampled the space and Shawn selected a well-spaced representative subset of a few thousand points.

IsoMap software [1] was used to initially explore the conformation samples, using non-linear dimension reduction. (See also the original IsoMap journal article [19] and website [18].) Each conformation has 72 dimensions, $3 \times 3 \times 8$, corresponding to the 3 positional and 3 rotational degrees of freedom for each of the 8 carbon molecules. IsoMap estimated the embedding dimension of the conformation space to be 5, using residuals of the non-linear projections. That is, the conformation space is a 2D object with an essential embedding dimension of 5, and which is given embedded in 72D. IsoMap generated a 3D visualization of it. Because the image is in 3D it is not apparent what is occurring in the remaining two dimensions. More specifically, in 3D there are apparently two rings of 'Y'-shaped intersections and a pinch point, but it is not obvious if these are actual intersections or just singularities of the projection.

So Shawn further analyzed the data using local projections and topology. First a triangulation was needed in order to have a complex on which to compute topology. Recall Shawn suspected that the space was locally two-dimensional, so we really mean triangles. Building a triangulation was difficult because existing surface reconstruction algorithms are limited to 3D, except for the incremental projection algorithm by Freedman [7]. So Shawn used Freedman's algorithm.

The algorithm initially failed in the 'Y'-shaped self-intersection neighborhoods. Exploring these regions, Shawn was able to discover that the data lay locally on two intersecting planes. The 'Y'-shape was an artifact of the projection: in 72D the intersections are 'X'-shaped. Shawn was able to locally partition the data and fit them to one of these two planes. Once that was done, the points on each plane could be triangulated separately using Freedman's algorithm. Care was taken so that the triangulations shared common edges near the planar self-intersections, and the triangles from the two planes could be put together without triangles interpenetrating, i.e. it was a well-defined simplicial complex with geometry. In this way the entire domain was eventually triangulated in 72D by Freedman's method [7]. The resulting triangulation was non-manifold (an edge in four triangles) in the neighborhood of these intersection regions, but was otherwise 2D and consistent.

No difficulties were encountered in the region of the apparent pinch point; it was purely an artifact of the non-linear projection.

Shawn computed the homology of the triangulation using both JPlex [3] and LinBox [12]. LinBox scaled better at the time. He got $\beta_0 = 1$, $\beta_1 = 1$ and $\beta_2 = 2$.

Shawn repeated the procedure for a variety of point subsamples and produced the same topological answers each time, which helped boost his confidence in the results.

Next the triangulation was further decomposed and analyzed topologically. The two rings of self-intersection were further analyzed. For one of the planes, in going around the ring, the plane connected to itself as in a cylinder. For the other plane, in going around the ring, the plane connected to itself as a Möbius strip. This may explain why the ring of intersection appeared as a 'Y'-shape instead of an 'X'-shape.

Ignoring the rings of self intersections, the variety decomposed into two closed surfaces. One surface contained both cylinders-around-the-ring and formed a sphere. The other surface contained

both Möbius strips and the apparent pinch point. It was a Klein bottle.

The two β_2 groups were analyzed. The Klein bottle cuts the interior of the sphere into two components: in the picture a donut about the equator; and the apple core of the north and south pole connected by the Klein bottle through the center of the sphere.

The one β_1 group comes from the Klein bottle.

β_0 is one since the sphere and Klein bottle are connected.

The crown conformations are around the north and south poles of the sphere.

Research Opportunities

The self-intersections proved to be both a blessing and a curse as they provided insight into where the mesh should be cut to decompose the structure, but were very difficult to triangulate.

Visual analytics were essential for designing the solution method for cyclo-octane. Visual analytics may not be sufficient for other, higher-dimensional, problems. As with cyclo-octane, in general conformations are varieties, and not necessarily manifolds. Is there a general solution methodology, say using algebraic geometry, that does not rely on visual analytics? Are there more automated methods for "triangulations" of molecular conformations with arbitrary dimensions and in arbitrary dimensions? Can the local dimension be determined automatically?

In general, geometrically-good homology generators would be useful for cutting a domain so that it could be non-linearly projected (e.g. using IsoMap) into more understandable regions. These would have been essential for this problem except the self-intersections served the same purpose.

Afra suggests computing homology over coefficients in $\mathbb{Z}_3$, as the torsion group of the Klein bottle should cause the result to be different than over $\mathbb{Z}_2$. Can this sort of test be automated for molecular conformations?

Converting topological structure into chemistry insight is a remaining challenge. At the very least topology must be translated into the language of chemists. Finding a Klein bottle in a conformation is exciting to a mathematician but (currently) meaningless to a chemist. Shawn has looked into what regions of the domain correspond to the crown, boat, and boat-chair conformations. A couple of graphs of the energy over the sphere and Klein bottle were produced. Mapping scalar quantities onto the resulting parameterization is a useful tool and may help chemists gain insight into their data.

In general chemists hope that molecules are ergodic. If $\beta_0 > 1$ then the molecule is clearly non-ergodic. Are there other conditions? Can these conditions be discovered or tested by topological tools?

Shawn Martin, Michael Kirby (Colorado State), Chris Peterson (Colorado State), Evangelos Coutsias, and Scott Mitchell wrote a proposal to further develop these dimensional reduction tech-

niques for algebraic sets.

3 Algorithms

The Sunday morning session focused on algorithmic challenges. The intended audience is those already familiar with the algorithm basics, rather than application owners. Speakers are asked to address one or more of the following aspects of their algorithmic approaches:

- algorithmic complexity, including dependence on genus, dimension, number of vertices, number of simplices, coefficient ring, filtration size, and number of critical points what's the hope for explicit bounds, tight bounds, and improvements?
- capabilities for sensitivity analysis and transient features, including Reeb graphs, filtrations, and new math structures
- application-tailored solutions, e.g. homology generators with specific geometry or cardinality; cycle homotopies
- visualization techniques that use topology

In addition, speakers were invited to comment on "visualization techniques for understanding topology" but this topic was not explicitly addressed.

See http://www.cs.sandia.gov/CSRI/Workshops/2009/CAT/program.html for links to some talks' slides.

3.1 Fast Vietoris-Rips Complex

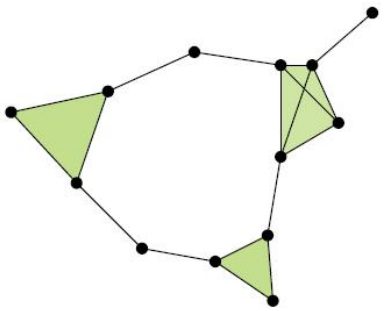

Figure 12. A V-R complex.

People

Afra Zomorodian (speaker). See also the last talk slide, "Acknowledgments."

Summary

The title of the talk was "Fast Construction of the Vietoris-Rips Complex." Trivia: Vietoris and his wife were very long lived. Rips is one of the original Bible code guys, but also does "real math." (Editorial comment: it is amusing but not-surprising that a pro-Bible-code website flips the emphasis, saying Rips is also a specialist "in the obscure micro-field of group theory.")

The idea is to more directly compute homology, bypassing constructions that are large and topologically uninteresting.

A typical problem starts with a point set sampled from an underlying space, and one seeks to recover the topology of the space. Typically the solution method is a two step process. In the first phase, a combinatorial representation of the data (the underlying space) is approximated (e.g. as a simplicial complex). The geometric information is abstracted away, leaving a combinatorial structure. In the second phase a topological invariant can be computed (e.g. persistent homology, β_i). People commonly mistake the first phase to be easier than the second phase; however the first phase is in fact much harder to do, and typically takes 99% of the time.

Geometric methods (e.g. apha complex) are fast but difficult to do, algebraic methods (e.g. Čech complex) are easy to do but are slow and produce huge complexes. Geometric methods such as alpha or flow complexes can be used. These are fast algorithms that generate embedded complexes. However the complexes are still large and require a Delaunay triangulation and are difficult to compute past 3 dimensions. Algebraic methods, including the Čech complex and Vietoris-Rips (VR) complex are simple, however they are slow to build and the results can get very large.

Talk slide 5 gives a nice one-slide summary of complexes.

This work focuses on decreasing the amount of time required to compute a VR complex. The speedup is from computing a VR neighborhood graph, then computing a VR expansion of it. The Vietoris-Rips complex is based on undirected edge weights $w(e_i)$, but Euclidean distances are not required. For a simplex σ, define $w(\sigma) = \max(w(\sigma_i) : \sigma_i \subset \sigma)$. You can sort these simplices by a weight function (maximum weight of the faces) to get a filtration ordering. The result is a weight-filtered complex. The VR complex is an expansion of the VR neighborhood graph. The computation of the neighborhood graph is a classic family of nearest or near neighbors problems, of which there are exact, ε-near, approximate, and randomized approaches. Tests were performed and it was determined that, in this setting, exact methods (kd-tree) should be used as there was no significant gain for going with a randomized approach.

Another technique is the witness complex, based on landmarks. Pick a subset of the vertices to be landmarks. Given two landmark vertices L_1 and L_2, then the edge between them is in the ε-witness graph if there is a third "witness" vertex v closer than ε to each of L_1 and L_2. Form the VR complex filtration ordered by ε.

An inductive algorithm was presented, based on the definition of the CW complex. Also two incremental algorithms and a maximal algorithm were presented. The exact algorithms are fast enough. The incremental algorithms are the fastest. In terms of timings, inductive interpolates

between incremental and maximal when in high dimensions.

Maximal clique is NP hard, and maximal clique is exponential size. So there is no point in doing approximate solutions.

VR complexes are great because they separate geometry from topology. Inductive methods of construction are intuitive, incremental approaches are natural for filtered data, and the maximal algorithm provides a minimum description in maximal simplices. Both steps of the maximal algorithm are parallelizable, and this is part of future work. Also, a current project computes both the complex and homology together in a single pass (half the time). You get a reduced collapsed set (around 3 orders of magnitude smaller), but there is no filtration on this set.

Research Opportunities

This is work in progress.

JPlex speed could be improved. Perhaps the chain complex could be built in one step. "Collapsed" and non-geometric, without a filtration, perhaps 2-level. Afra will follow up with the JPlex team.

Dmitriy Morozov mentioned that Zigzag persistence [2] does fast co-face (from a vertex) removal. Q: does Afra's new VR complex do that? A: Afra hasn't thought about it yet.

3.2 Linear Algebra for β and Torsions

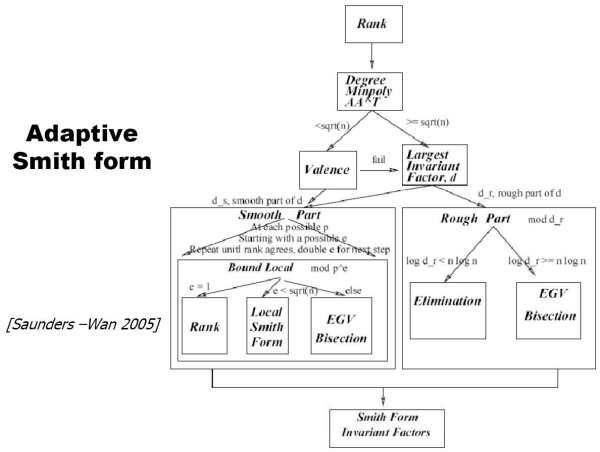

Figure 13. Flowchart for LinBox's adaptive Smith normal form.

People

Jean-Guillaume Dumas (speaker), also David Saunders, Erich Kaltofen, and Clement Pernet (speaker in Software session). See also the list of 38 people on the last talk slide, "Implementations."

Summary

The title of the talk was "Linear Algebra Algorithms for Betti Numbers and Torsions." A subtitle might be "LinBox algorithms: rank and Smith form using blackbox and sparse elimination methods." This talk described how Betti numbers can be determined via rank determination computations in the linear algebra package LinBox [12]. Two main methods were highlighted: a direct method, sparse elimination and reordering; and an iterative method, based on blackbox operations. A hybrid method combines them. The relationship between torsion and the coefficient ring was explored.

Direct Method

Sparse elimination (as in putting a matrix in Smith-Normal form) is a direct method. There is fill-in.

Reducing fill-in and coefficient growth are the main goals, as naive direct methods don't scale. A 13-node complete graph complex starts with a roughly 100,000 by 100,000 matrix with 800,000 non-zeros, and which takes only 3MB to store. But fill-in can give 10^{10} non-zeros. If one computes over $\mathbb{Z}$ the memory needed to store the coefficients grows as $2^{100,000}$. This example requires 150,000 GBytes.

In the beginning of the procedure, while getting the initial diagonal elements, you can control fill-in and avoid memory thrashing using intelligent pivoting heuristics. However, at the end of the procedure, you end up getting a ton of fill-in and you can run out of memory; see talk slides 6 and 8.

As with many elimination methods, choosing the order of pivot elements is key. For computation over small primes, it is better to recompute the ordering after each step of elimination. "Reordering" means choosing pivots to reduce fill-in. Finding a minimal reordering is NP-complete. However, there are reordering heuristics (from numerical methods) such as minimal degree ordering, and nested dissections. Reordering overhead is $O(n^2)$, where n is the number of rows and columns of the matrix.

One technique for controlling coefficient-size growth is to exploit the known factors of the diagonal terms. Symbolic factorization can be done first but you can end up with additional zeros.

Computer algebra using Gaussian elimination works on very over-determined systems. A proposed slightly intrusive heuristic is suited to small finite fields: in the sparsest row, take the sparsest

column. This is fast and proves effective. This is comparable to a method in superLU but superLU can require more memory.

Talk slide 11 gives experimental comparisons of the various strategies.

Iterative Method

Krylov Methods only use matrix-vector products and a full matrix is never modified; these are called blackbox. Krylov methods are iterative methods.

The Berlekamp/Massey algorithm is used. The matrix B is transformed to $D_1BD_2B^TD_1$. Next Weidmann's algorithm is applied to the transformed matrix to compute the minimal polynomial, which is the smallest relation between powers of the matrix. For random diagonal matrices, the eigenvalues of $D_1BD_2B^TD_1$ are distinct and its characteristic polynomial is x^k (minimal polynomial) with high probability. The degree of the minimal polynomial is the rank, with some probability, using Monte Carlo certifications.

Sparse elimination is typically faster than blackbox but cannot always provide an answer.

Hybrid Method

Introspective rank algorithms are hybrid direct and iterative methods. The introspective rank algorithm is basically a competition between elimination and black box. It eliminates until the blackbox estimate becomes faster. Block Wiedemann works well on large matrices.

Torsion

Torsion is computed using the Smith normal form. A naive implementation uses the greatest common divisors, and element growth is severe. Instead work with A^TA, and use the Valence Integer Smith form. The idea is to work modulo powers of prime numbers. This works if you know which primes will appear as factors of diagonal elements in the Smith normal form. These can be determined from the minimal polynomial. Another solution is to compute only the largest block invariant factor.

LinBox has an adaptive Smith normal form scheme. Splitting the problem into distinct problems over small primes decouples the problem; see talk slide 36.

These algorithms are implemented in the Gap homology package, LinBox and Sage. See the SimpHom package, in LinBox version 1.1.6, at http://www.linalg.org/download.html.

Research Opportunities

An open problem is to get fast certification of the rank, *a posteriori*, for iterative methods.

Future challenges include the need to be scalable; taking advantage of multicore / gpu architectures. Also efficient and reliable block algorithms should be defined. See additional challenges on talk slide 38.

Some discussions between the LinBox and Trilinos teams are taking place.

3.3 Short Loops

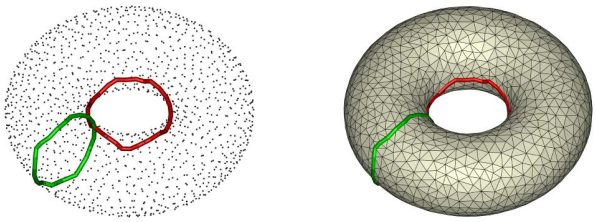

Figure 14. Shortest basis of $\mathsf{H}_1(M)$ for the torus.

People

Tamal Dey (speaker). Also co-authors Jian Sun (Princeton) and Yusu Wang (Ohio State). Jian was Dey's student, then Stanford post-doc, now Princeton post-doc.

Summary

The title of the talk was "Approximating Shortest Homology Loops." This work is not yet published elsewhere. Dey presented algorithms for computing the geometrically-shortest basis of H_1 for a (perhaps non-manifold) complex $\mathbb{K}$ in Euclidean space. Also, if instead of a complex one starts with a point-set P sampled from a manifold M, Dey's algorithms can compute an approximately geometrically-shortest basis of H_1 for M. Making the proofs was harder than developing the algorithms. The algorithms and proofs rely on geometric relationships between Čech and Vietoris-Rips complexes, point distances and filtration order, and M and Euclidean space. The coefficient ring $\mathbb{Z}_2$ is used. The algorithms are polynomial time in the simplices and β_1, but could be improved. Here $\beta_1 = \text{rank}(\mathsf{H}_1)$ is the first Betti number.

Algorithms

This work combines greedy characterization, persistence, and "intertwined complexes" as well as new observations. On the talks intro slide, last bullet, the three papers cited describe these building blocks.

This work is related to the localized homology algorithms of Zomorodian. A new algorithm by Freedman and Chen [4] for measuring homology classes and computing an optimal basis may be similar to this algorithm.

The goal is to look for 1-d cycles, loops. A set of loops generating H_1 is called a basis. Topological cycles can be computed from persistence algorithms, but in general these don't have good geometry. Geometry can be used to order the filtration, which is an indirect way of controlling the geometry of the cycles. A positive weight is associated with each edge, based on geometry, e.g. Euclidean distance. A weight is associated with each loop and the length of a set of loops is given by the sum of the weights. A shortest basis is the basis with minimal length (sum of weights of the edges).

It is assumed that the vertex set P of the point clouds being studied is a dense sample of a smooth manifold M. The Čech complex is defined in terms of the open balls B about the vertices. There is a nesting property between the Čech and Rips complexes; the Rips complex is nested between a two Čech complexes with different distance thresholds. The theorems rely on distance thresholds being carefully chosen based on the relationship between these complexes, and based on the geometric properties of the point sampling and the manifold.

The concept of geodesic is borrowed from differential geometry. A geodesic is a curve (path between two points) restricted to the manifold. Minimizing geodesics have minimum length. If the path endpoints are sufficiently close compared to the geometry of the manifold, the minimizing geodesic is unique. Geodesic and Euclidean distances can be linked if the points are sampled dense enough. This is related to the medial axis, and the feature size. The "reach," the minimum distance between M and its medial axis, comes into play.

The geodesic length may be used to define open balls, called geodesic balls. The convexity radius of a manifold (without boundary) at a point is the largest radius of a ball such that the shortest geodesic path between any two points lies entirely inside the ball. A Vietoris-Rips complex is formed, with distance threshold sandwiched between the sampling density and the convexity radius of the manifold. The first main theorem is that one may compute in polynomial time (about $O(\beta_1 n^4)$, where n = number of simplices) a set of loops on this complex that is an approximately shortest basis of $H_1(M)$.

The second main theorem is that a shortest basis for H_1 of any complex in an Euclidean space can be computed in $O(\beta_1 n^4)$, where n is the number of simplices. The algorithm for achieving this has three stages: greedy loops, canonical loops, and shortest loops. For greedy loops first the shortest generator is chosen, then the shortest generator for the remaining groups not spanned by the prior generators are chosen, etc. For canonical loops, the algorithm uses a filtration containing first the shortest path tree, then other edges ordered by the length of the *loop* they form when

added to the tree, then triangles. The persistent homology algorithm run on this filtration produces canonical loops. Another algorithm refines these canonical loops to shortest loops. The proof of correctness relies on defining simple cycles as "tight" if they contain a shortest path between every pair of points in the cycle, and showing that every loop in a shortest basis of $\mathsf{H}_1(\mathbb{K})$ is tight.

Research Opportunities

The proofs, particularly the connection to homology and the recent work of Smale, are subtle. The proofs may leave a little wiggle-room for adjusting the geometric conditions, but the geometric conditions appear to depend on one another and cascade throughout the arguments. It is possible that further study could reveal more relaxed or simpler geometric conditions.

Relatively little effort has been put into finding algorithms with optimal complexity. Most likely the complexity could be improved.

The proof is limited to computing homology over the coefficient ring $\mathbb{Z}_2$. One easy extension would be from $\mathbb{Z}_2$ to $\mathbb{R}$.

What about H_i for $i > 1$, especially 2 and 3. What about manifolds with boundary? What about non-Euclidean spaces?

Perhaps the following is an easier open problem: given a loop, can one find the shortest loop in the same homology class?

3.4 Embarrassingly Simple Reeb Graph Computation

People

Valerio Pascucci (speaker). [17]

Summary

The title of the talk was "Embarrassingly simple Reeb graphs computation." This talk describes modules in a larger system.

Reeb graphs are part of Morse theory. The context is functions on manifolds. A Reeb graph is a hierarchical data structure that encodes the structure of the level sets of the functions. Conceptually a Reeb graph is what remains when each isocontour component is contracted to a point; adjacent isocontours contract to adjacent points, forming paths. Some of the things that can be calculated are homology invariants, and some are dependent on the level set function. Reeb graphs have found many practical uses, e.g. matching shapes by matching graphs.

Previous methods that compute Reeb graphs trade generality for improved worst case complexity. For example, there are special algorithms for contour trees, which are defined only for simply connected domains. See also the Cole-McLaughlin et al. SoCG 2003 paper [5] for loops in Reeb graphs. This talk aims at generality and practical performance. The input model can be non-manifold and of any dimension. The algorithm sees the input triangles incrementally as in a streaming or out-of-core context. Some other algorithms rely on sorting the input triangles, say by height. But in these contexts one can't sort, and the algorithms were designed so there is no need to reorder the input. Another goal of this work was to build an algorithm that didn't hit the memory wall.

The algorithm presented to compute Reeb graphs is relatively simple: as new vertices, edges and triangles are added, update the Reeb graph accordingly. The algorithm is a kind of merge sort. Other work separates computation into stages; this is "dangerous" as one potentially computes lots of noise only to have to then reduce it, in which case most of the time is spent on the noise.

"Finalization" is the idea that you know when a vertex is used last, and won't appear again in the stream. I.e. the area around it is no longer actively changing, as all simplices containing it have already been seen. It is assumed that finalization information is provided. (If needed, you can perform a pre-pass through the data to collect this information.) The algorithm stores finalized areas more compactly in memory. Finalized information is removed from memory when computing Reeb graphs in out-of-core mode. There are a number of memory saving tactics employed, for example, input mesh triangles are never stored, finalized vertices are removed and edges with finalized vertices are removed. Also, Reeb graph arcs without edge references are retired as are nodes without edge references and adjacent arcs. By employing techniques like this, the amount of memory needed depends on the size of the largest level set as opposed to the size of the entire mesh.

The algorithm outline is to add a streamed triangle, check certain active level sets, update the Reeb graph, and throw away finalized data.

Simplification can be based on generalized persistence. But contraction based on geometry is often better than contraction based on persistence. Persistence, in the context of Reeb graphs, concerns the type of endpoints of an edge. Edges with weird endpoints are removed. Reeb graphs can be simplified by removing branches (extremum/saddle cancellation) and loops (saddle/saddle cancellation). Persistence of features is often defined in terms of differences in function values, however in some applications other measures may be more meaningful than regular persistence, e.g. medical imaging often uses volumes. There are some mixed results. This is subtle, and application specific. This is an active research area.

Simplifications can be used to develop shape signatures.

Only triangles are needed; it suffices to compute the Reeb graph from the 2-skeleton of the simplicial complex regardless of the dimension of the original mesh. This is because the Reeb graph is a 1-skeleton object. The Sierpinski triangle (or gasket or sieve) is an example of a fractal mesh of arbitrary dimension, with complex topology.

This algorithm computes Reeb graphs of any dimension, however when working in arbitrary dimensions you can get many simplices and highly non-manifold structures. So, although it is robust it is computationally more expensive. However practical tests confirm robustness and show good scalability; in 2d the worst-case complexity is $O(n^2)$. The order in which the simplices stream by affects the running time and memory requirements.

Research Opportunities

An implementation is currently being ported to VTK [20].

Parallel algorithms for Reeb graphs is a challenge.

The community would like a corpus of datasets. Perhaps we could make a Sandia repository, starting with the CAT workshop Application session datasets. Perhaps Valerio could host a repository at Utah. There are issues with dataset ownership and data size that could complicate things.

3.5 Computational Homology Project (CHomP)

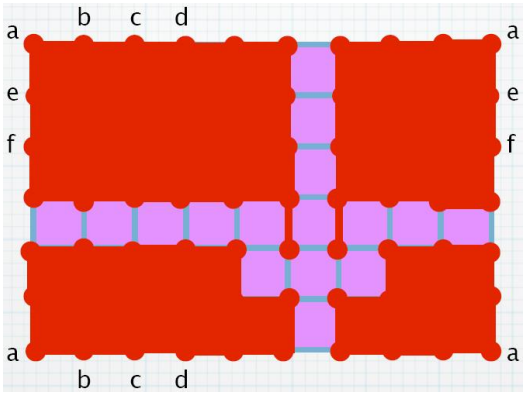

Figure 15. A large acyclic set for a cubical mesh of a torus used for coreduction.

People

Konstantin Mischaikow (speaker). See also Thomas Wanner's talk in the Software session. Computational Homology [10] book with Tomasz Kaczynski(U. Sherbrooke), Konstantin, and Marian Mrozek (Jagiellonian University).

Summary

Konstantin prefaced his talk by saying this was "a different set of topics" than the rest of the workshop. Understanding maps is really what it is all about, the goal is not simply β_i. See also his comments in the Panel discussion.

The way that you understand an object is by understanding its morphisms, the set of all maps. Restricting to certain maps and spaces helps, e.g. linear maps, vector spaces.

The mechanics are to work with free chain complexes consisting of free abelian groups, called chains; and homomorphisms, called boundary operators. Homology, namely Betti numbers, are computed using Smith normal form.

Geometric Complexes

Cubical complexes are used, whose elements are finite products of elementary intervals. Element $Q = I_1 \times I_2 \times ... \times I_d \subset \mathbb{R}^d$ where interval $I = [l, l+1] \text{or} [l, l] \subset \mathbb{R}$. Elements of dimension d in a cubical complex have 2^d points, and are combinatorial points, edges, squares, hexahedra, etc. Some data come in the form of a bit map that has been thresholded, which are naturally cubical. Cubical meshes are also a very efficient way to bin datapoints or subdivide continuous spaces. Efficiency is critical when datasets are large.

Co-reduction

Co-reduction is used to reduce a geometrical complex as much as possible prior to building the chain complex and computing the Smith-normal form. The co-reduction algorithm of Mrozek and Batko [14] is efficient. The co-reductiomn algorithm finds a maximal acyclic subset. At its core, it is a breadth first search. Cubes are grown outward as much as is possible without changing topology, producing an acyclic subset. For homology, these acyclic subsets may be quotiented out or ignored by relative homology.

The goal is to pick a big subset a, which takes out a lot of cycles. The downside is that geometry is lost. If $a \subset x$ is acyclic, then you can write the homology of x in terms of x relative to a. The goal then is to remove a maximal acyclic subset a of x. This results in computing a local geometric boundary and co-boundary which are implemented as bitmaps.

When working with random cubicle complexes of around 10^5 elements in $\mathbb{R}^4$ with Betti numbers around 10^4 this process takes only a few seconds. An example problem with 350×10^6 simplices in $\mathbb{R}^4$ takes about 15 minutes.

Maps

However, when working with maps this process is much slower. Given two chain complexes, if you want to compute a map between these you need to commute the map with the boundary operators. This process is well defined if boundaries go to boundaries.

When working with cubicle structures then you have chain maps that take you from geometry to algebra. But note that geometric reductions require lots of book keeping.

Dynamical Systems

Understanding dynamical systems is an important example. Integrating a differential equation maps a space to itself over time. The map is applied to a set of starting points, any cube that a mapped point lands in is marked as being part of the range. (Mapping points to cubes rather than points is an effective strategy for doing quick calculations while tolerating relatively large numerical errors.) Map bifurcations are observed by computing the topology of the cubes of the range.

It is extremely expensive to examine the whole phase space. Alternatively, consider a parameterized map, such as the logistic map. It is expensive to find good estimates of a parameter. This procedure helps do that efficiently.

Research Opportunities

The software will be delivered next spring (2010). The software only works for cubical complexes. Software for simplicial complexes is under development, with planned release next fall (2010). Work on chain maps is in progress.

CHomP website is http://chomp.rutgers.edu/ [13]. Related material is available from the Computer Assisted Proofs in Dynamics web site, http://capd.wsb-nlu.edu.pl [15].

4 Panel

4.1 People

Panelists: David Saunders, Peer-Timo Bremer, Dmitriy Morozov, Shawn Martin, and Michael Robinson.

Moderator: Scott Mitchell.

Lively contributions by Konstantin Mischaikow, Afra Zomorodian, Tamal Dey, Valerio Pascucci ...

4.2 Questions

Q1. Rank applications in terms of low hanging fruit that topology can pluck.

Q2. What are the key open problems, or main roadblocks, for advancing algorithms?
In particular, comment on scalability, techniques for high dimensional data, and generalizations of filtrations

Q3. What new software or software mechanisms/structures would most benefit the community?
"New software" means, what techniques would be valuable to have in accessible and general purpose format such as LinBox and Plex? "Mechanisms/structures" means, would an open source effort be helpful? And should general and available versions be developed of Reeb graphs, complex generation methods, or anything else in particular?

4.3 Panelists' Answers

Peer-Timo Bremer: Q1, Q2. We should be performing the same analysis as scientists; however we should do it for all threshold values, e.g. compute merge tree, give scientists ability to compute

conditional statistics across all threshold values. Topology can be used for data compression. Rather than write out complete state, topology can be written out. We need to put more effort into making things work in practice as opposed to just identifying the theory. (Q3.) Also, there are tradeoffs between a specialized code vs. a general templated library. General libraries can be slower but may have much wider applicability.

David Saunders: Q1. The persistent homology problem of incremental rank is a problem that faces many communities: cryptography and polynomial systems are two examples. While it is not necessarily low hanging fruit, speeding things up can be done using block updates. It would be very easy to add torsion. Torsion for low p is not that hard to determine, as mentioned by Jean-Guillaume Dumas.

Dmitriy Morozov: Q2, Q3. Two approaches to data analysis using topology are the study of scalar fields and the homology of point clouds. The main obstacle to applying these approaches is a lack of data structures for complexes (Michael Robinson agreed). Large challenges include developing multi-dimensional persistence and generalizing persistence. Existing algorithms need to be speeded up.

As to what is needed for software, it is too early for open source repositories. Computational topology is a very young field and not yet stable. This makes it hard to define interfaces, for example. Probably the best the community can do at this point is make our individual codes available on our websites, and wait before trying to develop a collective product.

Shawn Martin: Q1. A lot of the low hanging fruit has already been plucked. Sensor networks and finite element mesh data analysis and visualization seem like low hanging fruit, while text analysis and analysis of non-manifold structures is much harder. For most settings, other than what Tamal Dey presented, most of the software doesn't produce generators. Even then, rather than just computing loops, we should be looking at higher dimensional generators, e.g. voids. It would be great if the community starting posting their codes to websites but Shawn agreed with Dmitriy that it is too early to try to define a community-wide open software effort.

Michael Robinson: Q1. A theme is "how do you take large, high-dimensional data and get it down to something that is workable?" A lot of this filtering is reminiscent of signal processing. (Define "topological signal processing" as a new field?) Many people are working with very elementary topological tools. What would happen if you worked with something that provided a bit more structure, such as different coefficient rings? We need to add good local structure and this may be doable by building local algebraic information from discrete sheaves, a topic in sheaf cohomology. Understanding sheaves might be a powerful tool for searching through large data sets, identifying regions to ignore. There are situations, in plasma physics for example, where the knots in the loop groups are important. These types of features can only be resolved using homotopy, which is much harder than sheaf cohomology. When considering computational homology and homotopy classes, you get a map between two spaces. What form should the data take, e.g. simplices, cubes, minimal structures? Tamal and Afra commented that working with homotopy & homeomorphy becomes intractable above 4D.

4.4 Further Discussion and Summary

Maps. Konstantin was surprised by the starting point for many of the talks in the workshop. Many talks started with an embedded point set, some with a complex. Workshop participants were encouraged to remember that homology is all about maps. A possible explanation for why this is relatively unexplored is because people who are doing applications do not think of homology in terms of maps. Afra commented that as a CS community we are often trained to think in terms of data and datastructures, and not about the maps between data. (Functional language theory?) Maps are often there in the original application, but standard computer science procedures often discard them. Persistence is homology of a certain map.

Q1. Applications. How can we get from initial data to the point of being able to compute homology? If the complex is evident, the problem is easier. When a complex is not so clear, or comes from an algebraic variety (with non-manifold self-intersections) as in Shawn's talk, the problem is harder. Finite-elements are an intermediate-difficulty data type.

What do Betti numbers even mean in some situations?

Q2. Open problems. A big open problem is to do homology for multiple parameters or thresholds at once. This would enable computing conditional statistics, for example. This problem is addressed to some degree by the zigzag persistence methods [2] for homology, which are a recent advance by Dmitriy et al.

In persistent homology, the linear algebra does not seem to be a bottleneck. However, the problem of incremental rank is important in this and other fields.

The complexes that we are working with are hard to represent. Better data structures are needed. For a fixed number of points, the complex size blows-up as the dimension increases. Is there a way to finesse the curse of dimensionality?

Non-manifold structures are an open problem in some settings.

Q3. Software. Challenges are to make code work in practice, with large data sets, and reliably. There is a conflict between speed and reliability. Different codes are needed at different extremes for different uses.

It seems like a lot of workshop participants have their own private codes. It would help to share software more.

It would be helpful to have a library of models, datasets to run software on and compare results. Valerio would like a core data repository for high-dimensional data so that methods can be tested against "real" data.

4.5 Editorial Comments on Algorithms and Software

There appears to be considerable opportunity to write formal bounds for the complexities of homology algorithms, especially their dependence on various interrelated aspects such as the number of points, number of simplices, and dimension of simplices; value of Betti numbers, and generating cycle length; whether generators are computed or just ranks; also coefficient size for arithmetic over rings and rationals. Some researchers have internal confidence that they understand the complexity of various algorithms, but this has not been written down in a form digestible by the community. It seems that most people focus on making the algorithms faster, rather than understanding the current speed.

The algorithms themselves are in flux, and the publicly available codes for homology in general dimensions do not provide generators or parallel computation. There are generators available for 3d solid modeling applications, and geometrically good generators are more well developed in that area.

The basic paradigm of computation appears to be three separate stages: generate a complex, perform reduction, compute homology. It would be valuable to have a formal study into a more integrated view of the process, and a full exploration of ideas such as suggested by Afra Zomorodian in his talk about combining these steps so as not to compute lots of simplices that have null homology and contribute nothing. These could provide significant improvements to algorithmic complexity, and a formal analysis of these integrated processes would be valuable.

5 Conclusion

The workshop was a great success. In keeping with a workshop, the talks were able to focus on unsolved problems, very recent results, and letting other people know about activities and capabilities. It got a critical mass of people together to talk, some for the first time with each other. People made plans for further collaboration. It would be worthwhile to have more combinatorial algebraic topology (CAT) workshops in future years.

References

[1] William "Mike" Brown, Shawn Martin, Haixia Jia, and Jean-Paul Watson. Dimensionality reduction library (Dr. L). http://www.cs.sandia.gov/~wmbrown/drl/index.htm.

[2] Gunnar Carlsson, Vin de Silva, and Dmitriy Morozov. Zigzag persistent homology and real-valued functions. In *SCG '09: Proceedings of the 25th annual symposium on Computational geometry*, pages 247–256, New York, NY, USA, 2009. ACM.

[3] Gunnar Carlsson et al. Topological methods in scientific computing, statistics and computer science, JPlex project web site. http://comptop.stanford.edu/.

[4] Chao Chen and Daniel Freedman. Measuring and computing natural generators for homology groups. *Comput. Geom. Theory Appl.*, 43(2):169–181, 2010.

[5] K. Cole-McLaughlin, H. Edelsbrunner, J. Harer, V. Natarajan, and V. Pascucci. Loops in Reeb graphs of 2-manifolds. In *Proc. 19th ACM Symposium on Computational Geometry (SoCG)*, pages 344–350, 2003.

[6] Jean-Guillaume Dumas, Frank Heckenbach, David Saunders, and Volkmar Welker. Computing simplicial homology based on efficient smith normal form algorithms. *Algebra, Geometry, and Software Systems*, pages 177–206, 2003.

[7] Daniel Freedman. An incremental algorithm for reconstruction of surfaces of arbitrary codimension. *Comput. Geom. Theory Appl.*, 36(2):106–116, 2007.

[8] Geomagic. Geomagic commercial website. http://www.geomagic.com/.

[9] Michael Heroux et al. The Trilinos project website. http://trilinos.sandia.gov/.

[10] Tomasz Kaczynski, Konstantin Mischaikow, and Marian Mrozek. *Computational Homology (Applied Mathematical Sciences)*. Springer, 1st edition, January 2004.

[11] Kitware. Titan website. https://www.kitware.com/InfovisWiki/index.php/Main_Page.

[12] LinBox. Project LinBox: Exact computational linear algebra website, linbox-use@googlegroups.com. http://linalg.org.

[13] Konstantin Mischaikow et al. CHoMP, computational homology project homepage. http://chomp.rutgers.edu/.

[14] Marian Mrozek and Bogdan Batko. Coreduction homology algorithm. *Discrete & Computational Geometry*, 41(1):96–118, 2009.

[15] Marian Mrozek et al. Computer assisted proofs in dynamics website. http://capd.wsb-nlu.edu.pl.

[16] M Ortiz and A Pandolfi. Finite-deformation irreversible cohesive elements for three-dimensional crack-propagation analysis. *Int. J. Numer. Meth. Eng.*, 44(9):1267–1282, 1999.

[17] Valerio Pascucci. Homepage. http://pascucci.org/.

[18] J. B. Tenenbaum. A global geometric framework for nonlinear dimensionality reduction. Website. http://waldron.stanford.edu/~isomap/.

[19] J. B. Tenenbaum, V. de Silva, and J. C. Langford. A global geometric framework for nonlinear dimensionality reduction. *Science*, 290(5500):2319–2323, December 2000.

[20] VTK. Visualization toolkit, VTK, website. http://www.vtk.org/.

[21] Bernadette Watts, Deanna Ceballos, and Scott Mitchell. Combinatorial algebraic topology (CAT): Software, applications & algorithms, CSRI workshop website. http://www.cs.sandia.gov/CSRI/Workshops/2009/CAT/index.html.

[22] D. H. Wiedemann. Solving sparse linear equations over finite fields. *IEEE Trans. Information Theory*, IT-32:54–62, 1986.

[23] Afra Zomorodian and Gunnar Carlsson. Computing persistent homology. *Discrete Comput. Geom.*, 33(2):249–274, 2005.

DISTRIBUTION:

1	MS 9159	Janine Bennett, 08963
1	MS 1320	Scott Collis, 01416
1	MS 1320	David Day, 01414
1	MS 1318	Bruce Hendrickson, 01410
1	MS 1316	Shawn Martin, 01415
7	MS 1318	Scott Mitchell, 01415
1	MS 1323	David Rogers, 01424
1	MS 1318	Suzanne Rountree, 01415
1	MS 0899	Technical Library, 9536 (electronic)

Sandia National Laboratories